

IN THE CLAIMS:

Claims 1-19 were previously cancelled without prejudice or disclaimer of the subject matter recited therein. Please amend claims 20, 32, and 43 as follows:

Claims 1-19 (Cancelled)

20. (Currently Amended) A method of implementing an application and eliminating uncontrolled internal interdependencies within the application, with the application comprising a number of functional entities, each entity comprising one or more elements, and with the application producing application output data from application input data such that element output data produced by the elements determines entity output data produced by the functional entities and the entity output data determines the application output data, wherein there are interdependencies formed between the elements, between the functional entities, or between the elements and the functional entities, said method comprising:

normalizing an element such that uncontrolled internal interdependencies within said element are eliminated, ~~and such that the input data supplied to said element solely determines element output data produced by said element.~~ wherein said normalizing comprises the steps of:

dividing the application input data into parts small enough for each element input data to solely determine the corresponding element output data,

analyzing interdependencies between each entity input data and the corresponding entity output data,

performing a search for such combinations of entity input data which solely determine the entity output data, and

forming a normalized element out of the found combinations, such that the element input data solely determines the element output data.

21. (Previously Presented) A method as recited in claim 20, wherein at least one normalized element forms a functional entity, and wherein output data of the functional entity is determined by the at least one normalized element.

22. (Previously Presented) A method as recited in claim 20, further comprising:
forming at least one functional entity from at least one normalized element; and
normalizing the at least one functional entity.

23. (Previously Presented) A method as recited in claim 20, wherein the normalized element comprises a normalized functional entity.

24. (Previously Presented) A method as recited in claim 20, wherein the normalized element forms at least one normalized functional entity, and wherein the output data of the application is solely determined by the normalized functional entity.

25. (Previously Presented) A method as recited in claim 20, wherein said normalized element forms at least one normalized functional entity, and wherein the at least one normalized functional entity is formed by joining at least two elements of said plurality of elements in parallel, as a union.

26. (Previously Presented) A method as recited in claim 20, wherein said normalized element forms at least one normalized functional entity, and wherein the at

least one normalized functional entity is formed by selecting at least one item of element output data.

27. (Previously Presented) A method as recited in claim 20, wherein said normalized element forms at least one normalized functional entity, and wherein the at least one normalized functional entity is formed by specifying element output data based upon input data.

28. (Previously Presented) A method as recited in claim 20, said method further comprising the steps of:

executing a plurality of normalized elements in cascade, wherein the output data of the element affects input data of another element of the plurality of normalized elements.

29. (Previously Presented) A method as recited in claim 20, further comprising the steps of:

conditionally executing normalized elements, wherein when a second normalized element produces predetermined output data, input data for a first normalized element is determined, and wherein when the second normalized element does not produce the predetermined output data, no input data is determined for the first normalized element.

30. (Previously Presented) A method as recited in claim 20, further comprising the steps of:

repeatedly executing the normalized element, with a number of repeated executions of a first element being determined by output data of a second element.

31. (Previously Presented) A method as recited in claim 20, wherein the application is implemented as telephone exchange software in a mobile communication system.

32. (Currently Amended) A system for implementing an application and eliminating uncontrolled internal interdependencies within the application, wherein the application comprises a number of functional entities, each entity comprising one or more elements, and wherein the application produces application output data from input data such that element output data produced by the elements determines entity output data produced by the functional entities and the entity output data determines the application output data, and wherein there are interdependencies formed between the elements, between the functional entities, or between the elements and the functional entities, said system comprising:

normalizing means for normalizing one or more elements such that uncontrolled internal interdependencies within each element are eliminated, ~~and such that the input data supplied to each element solely determines element output data produced by said element~~ wherein said normalizing means comprises:

dividing means for dividing the application input data into parts small enough for each element input data to solely determine the corresponding element output data,

analyzing means for analyzing interdependencies between each entity input data and the corresponding entity output data,

performing means for performing a search for such combinations of entity

input data which solely determine the entity output data, and

forming means for forming a normalized element out of the found combinations, such that the element input data solely determines the element output data.

33. (Previously Presented) A system as recited in claim 32, further comprising:
forming means for forming a functional entity comprising at least one normalized element.

34. (Previously Presented) A system as recited in claim 33, further comprising:
normalizing means for normalizing the functional entity.

35. (Previously Presented) A system as recited in claim 34, further comprising:
normalizing means for treating the normalized functional entity as a normalized element.

36. (Previously Presented) A system as recited in claim 32, further comprising:
forming means for forming an application comprising at least one normalized functional entity, and wherein the output data of the application is solely determined by the at least one normalized functional entity.

37. (Previously Presented) A system as recited in claim 32, further comprising:
forming means for forming a functional entity from at least one normalized element, said forming means joining at least two elements of said plurality of elements in parallel, as a union.

38. (Previously Presented) A system as recited in claim 32, further comprising:

at least one normalized functional entity, and wherein the at least one normalized functional entity is formed by selecting at least one item of element output data.

39. (Previously Presented) A system as recited in claim 32, further comprising:
at least one normalized functional entity, and wherein the at least one normalized functional entity is formed by specifying element output data based upon input data.

40. (Previously Presented) A system as recited in claim 32, further comprising:
executing means for executing a plurality of normalized elements in cascade,
wherein the output data of the element affects input data of another element of the plurality of normalized elements.

41. (Previously Presented) A system as recited in claim 32, further comprising:
forming means for forming a functional entity based upon at least one normalized element, wherein said at least one normalized element can be executed conditionally, and wherein when a second normalized element produces predetermined output data, input data for a first normalized element is determined, and when the second normalized element does not produce predetermined output data, no input data is determined for the first element.

42. (Previously Presented) A system as recited in claim 32, further comprising:
forming means for forming a functional entity from at least one normalized element, wherein the at least one normalized element can be executed repeatedly, with a number of repeated executions of a first element being determined by output data of a second element.

43. (Currently Amended) A system for implementing an application and eliminating uncontrolled internal interdependencies within the application, wherein the application comprises a number of functional assemblies, each assembly comprising one or more elements, and wherein the application produces application output data from input data such that element output data produced by the elements determines assembly output data produced by the functional assemblies and the assembly output data determines the application output data, and wherein there are interdependencies formed in at least one of between the elements, between the functional assemblies, and between the elements and the functional assemblies, said system comprising:

the one or more elements; and

normalizing means for normalizing one or more elements such that uncontrolled internal interdependencies within each element are eliminated, ~~and such that the input data supplied to each element solely determines element output data produced by said element~~ wherein said normalizing means comprises:

dividing means for dividing the application input data into parts small enough for each element input data to solely determine the corresponding element output data,

analyzing means for analyzing interdependencies between each entity input data and the corresponding entity output data,

performing means for performing a search for such combinations of entity input data which solely determine the entity output data, and

forming means for forming a normalized element out of the found combinations, such that the element input data solely determines the element output data.